Energy Systems
https://doi.org/10.1007/s12667-021-00477-1

ORIGINAL PAPER

A teaching–learning-based optimization algorithm for the environmental prize-collecting vehicle routing problem

Dimitra Trachanatzi[1] · Manousos Rigakis[1] · Magdalene Marinaki[1] · Yannis Marinakis[1]

Received: 25 March 2020 / Accepted: 18 August 2021

Abstract

The present research proposes a new Vehicle Routing Problem (VRP) variant, the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP). According to the original PCVRP formulation, the scope of the problem is to maximize the total collected prize from the visited nodes and simultaneously minimize the fixed vehicle-utilization cost and the variable cost. In the E-PCVRP formulation, the variable cost is not solely expressed as a vehicle-covered distance but as a load-distance function for CO_2 emissions minimization. The Teaching–Learning-Based Optimization (TLBO) algorithm is selected as the solution approach. However, TLBO is designed to address continuous optimization problems, while the solution of the E-PCVRP requires a discrete-numbered representation. Thus, a heuristic encoding/decoding technique is proposed to map the solution in a continuous domain, i.e., the Cartesian space, and transform it back to the original form after applying the learning mechanisms, utilizing the Euclidean Distance. The encoding/decoding process is denoted as CRE, and it has been incorporated into the standard TLBO algorithmic scheme, and as such, the proposed TLBO-CRE algorithmic solution approach emerges. The effectiveness of the TLBO-CRE is demonstrated over computational experiments and statistical analysis in comparison to the performance of other bio-inspired algorithms and a mathematical solver.

Keywords Teaching–learning-based optimization algorithm · Prize-collecting vehicle routing problem · Environmental vehicle routing problem · Carbon emissions minimization

✉ Dimitra Trachanatzi
dtrachanatzi@isc.tuc.gr

Extended author information available on the last page of the article

1 Introduction

According to the European Environment Agency (EEA), heavy-duty vehicles, such as lorries, buses, and coaches, are responsible for about a quarter of CO_2 emissions from road transport in the European Union (EU), and for 6% of total CO_2 emissions in the EU [1]. Despite some improvements in fuel consumption efficiency in recent years, these numbers are still rising, while as set out in the 2011 Transport White Paper [2] emissions need to fall by around two thirds by 2050, compared with 1990 levels, in order to meet the long-term 60% greenhouse gas emission reduction target. As such, carriers and logistics operators need to effectively and efficiently design eco-friendly supply chains and, consecutively, consider the carbon emissions in their transportation scheduling. That environmental aspect should also be considered in the optimal design of vehicle routes. Hence, several academic studies are now focusing on green variants of the classical Vehicle Routing Problem (VRP), as the effect of the routing on the environment is to be minimized [3]. The VRP is an NP-hard combinatorial optimization problem that considers the assignment and sequencing of nodes (clients) to vehicle routes, based on several imposed constraints, while optimality is achieved in terms of cost minimization, where cost may be related to financial costs, distance, time, environmental cost, penalty, and others [4].

The Vehicle Routing Problem with Profits (VRPPs) is a VRP class, where the set of nodes to be visited is not pre-specified, and each one is associated with a prize value. Such a selection of the most profitable ones should occur [5]. Thus, the objective is to minimize the difference between the total collected prizes and the total routing costs. VRPPs have been used to model real-life problems and applications in a variety of areas, and one of the problems that fall in this class is the Prize-Collecting VRP (PCVRP) [6]. The main scope of the problem is that for a given set of nodes (each one associated with a demand and a prize value), a pre-specified number of routes should be formed to visit them, taking into account both the total collected prize maximization, the cost minimization, as the total traveled distance and the fixed vehicle utilization cost. The characteristics that distinguish the PCVRP from the original VRP is that: (1) the size of the utilized homogeneous fleet of vehicles is not sufficient to cover the demand of all nodes, and such, it is not compulsorily required for every node to be visited and (2) the total demand of the visited nodes should not be less than a pre-specified value, i.e., service level requirement. The present research expands the formulation of the PCVRP, introducing the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP). In the proposed E-PCVRP, the cost minimization objective, as the total distance traveled, is replaced by a load-distance function for CO_2 emissions minimization.

The E-PCVRP, as a VRP variant, is an NP-hard problem, making the search of optimal solutions a computationally demanding task, and such, striving for computational efficiency, a heuristic-based approach should be utilized in its solution. In the present research, the Teaching–Learning-Based Optimization (TLBO) algorithm is selected as the basis of the proposed solution approach. TLBO is a bio-inspired meta-heuristic algorithm that offers competitive convergence and accuracy characteristics. As a stochastic search scheme, TLBO is characterized by a simple framework, without algorithm-specific parameters, and easy implementation. Initially, TLBO has been proposed to obtain global solutions for continuous non-linear functions in mechanical design optimization [7]. However, the proposed E-PCVRP requires discrete-numbered solution encoding to represent the visit-sequence of nodes. In order to avoid the discretization or hybridization of the standard algorithm TLBO algorithmic scheme, a novel encoding/decoding process is presented, namely the CRE. According to CRE, a discrete VRP solution representation is transformed into continuous, utilizing the Cartesian coordinates of each node. The transformation is data-based and corresponds to the original movement process, omitting the necessity to alter the standard algorithmic framework and eliminating approximation or transformation errors. For this research, the CRE was embedded into the TLBO algorithmic scheme, forming the proposed TLBO-CRE. Inspired by the open challenges in the field of bio-inspired computation that Osaba et al. [8] highlighted, the scope of the present research is not solely to introduce another variant of a known bio-inspired algorithm but to utilize its intelligent bio-inspired processes for the solution of a discrete optimization problem, preserving the respective standard algorithmic scheme. Thus, the contribution of this paper is twofold, the introduction of the E-PCVRP as a novel VRP variant, including prize maximization and environmental cost minimization, i.e., CO_2 emissions minimization, and the proposal of the CRE encoding/decoding process. The main advantage of the CRE is that it could be incorporated into any bio-inspired algorithm designed for continuous optimization for the solution of any given VRP. Since the encoding process relies solely on the position of the nodes, the decoding process could be altered to fit the constraints imposed by the formulation of any VRP. The rest of the paper is organized as follows: in Sect. 2 a brief literature review on the PCVRP and on the TLBO are presented; in Sect. 3 the mathematical formulation of the PCVRP is introduced, while in Sect. 4 the proposed formulation of the E-PCVRP is given; in Sect. 5 the original TLBO framework is briefly described; in Sect. 6 the proposed TLBO-CRE is presented; in Sect. 7 the computational experiments and the statistical analysis are reported; and finally, the drawn conclusions are given in the last section.

2 Brief literature review

2.1 Literature related to the prize-collecting vehicle routing problem

The Prize-Collecting Vehicle Routing Problem (PCVRP), has been proposed, in 2006, by Tang and Wang [6] for the solution of the hot rolling production scheduling problem, utilizing an Iterated Local Search algorithm (ILS) based on Very Large-Scale Neighborhood (VLSN). The objective in the mathematical formulation of the PCVRP is a linear weighted combination of three others: the minimization of the total traveled distance, the minimization of the number of utilized vehicles (along with their associated fixed usage cost), and the maximization of the total collected prize. In the same field, Zhang et al. [9] proposed a multi-objective PCVRP-based model, in which penalty is associated with unvisited nodes and solved via a variant of the Particle Swarm Optimization (PSO) algorithm. The aforementioned problem, with a similar mathematical formulation, was solved by Jia et al. [10] using a Pareto Max-Min Ant System (P-MMAS). In 2015, Tiwari et al. [11] proposed a Hybrid Edge Recombination approach (HER) for the solution of a PCVRP formulation, incorporating penalty values in the objective function. In 2016, Li and Tian [12] aggregated the three aforementioned PCVRP objectives in one function and presented a Two-level Self-Adaptive Variable Neighborhood Search algorithm (TLSAVNS) to solve the problem. Two variants of the PCVRP derive from the literature, the PCVRP-P with a pre-specified number of vehicles to be utilized and the PCVRP-NP with a variable number of vehicles. In many instances, the problem is addressed as a single-objective optimization problem, employing the weighted-sum method, in which objectives are aggregated by assigning a deterministic weight to each one. In 2019, Long et al. [13] considered both variants of the PCVRP as multi-objective problems, using a Pareto-based evolutionary algorithm for the PCVRP-P, i.e., a hybrid multi-objective genetic local search (HGLS) algorithm featuring local search strategy and a decomposition strategy for the PCVRP-NP. Several variants of the PCVRP have been proposed in the literature, including penalty functions, multiple depots, and various service level constraints. Stenger et al. [14] presented a PCVRP-based formulation with non-linear cost to model a small package shipping problem and developed an Adaptive Variable Neighborhood Search (AVNS) algorithm to solve it. The VRP with service levels (VRP-SL) has been proposed by Bulhoes et al. [15], as an extension of the PCVRP to multiple groups, from the viewpoint of a third-party logistics provider, solved via a branch-and-price algorithm and a first effective hybrid genetic search. Recently, in 2019, Orlis et al. [16] presented the Capacitated Routing Problem with Profits and Service Level Requirements (CRPPSLR) inspired by the operations of Cash-In-Transit (CIT) company and utilized a branch-and-cut algorithm to solve it. The proposed PCVRP variant, namely the E-PCVRP, includes the CO_2 emissions minimization objective, which takes into account the amount of cargo that a vehicle has to carry over a corresponding distance. At least to our knowledge, there is no work found in the literature that proposes a PCVRP variant incorporating an environmental aspect, such as CO_2 emissions minimization.

2.2 Literature related to the teaching–learning-based optimization algorithm

Teaching–Learning-Based Optimization (TLBO) algorithm is a population-based heuristic optimization algorithm, which does not require algorithm-specific parameters and provides rapid convergence and easy implementation. As reported in recent surveys [17–19], it has been successfully employed to solve numerous problems from diverse scientific fields, such as, manufacturing and operations research [20], mechanical and electrical engineering [21], civil engineering [22, 23], data clustering [24] and others. Moreover, in many studies, the original TLBO algorithmic scheme has been extended and modified to enhance its exploration and exploitation abilities and avoid falling into local optimum. For example, the TLBO has been enhanced by incorporating initialization techniques [25], adaptive parameters [26], learning strategies [27], population neighborhood [28] and it has been hybridized with other search techniques [29] and optimization algorithms [30]. Based on the related literature TLBO algorithm performs excellently in dealing with continuous problems; however, there is little research on the TLBO algorithm in dealing with discrete problems [19]. Studies found in the literature employ discretized versions of the TLBO algorithm to address discrete-optimization problems. In 2014, Dede [22] utilized the TLBO on the weight minimization of truss structures with discrete design variables. A discrete version of the TLBO (DTLBO) has been presented by Li et al. [31] for solving the flowshop rescheduling problem. In their proposed approach, two types of heuristics for both teaching and learning phase of the standard algorithmic scheme, are presented. Concerning the discretization approaches, Li et al. utilized simple heuristic operators, and transformation equations, while in the decoding process, each element (float number) is transformed back into an integer by taking the integer part of it. Another discrete TLBO has been proposed by Lotfipour and Afrakhte [32], in 2016, to solve the distribution system reconfiguration problem, utilizing mapping equations. Chen et al. [33] presented one more DTLBO, for optimizing community detection problems, according to which learners are coded by real integer values, and the updating rules for learners are redesigned to incorporate simple operators, e.g., XOR, niching mechanism and mutation operation. Shao et al. [34, 35] presented a hybrid discrete optimization algorithm based on teaching-probabilistic learning mechanism (HDTPL) to solve the no-wait flow shop scheduling with minimization of makespan, and the no-idle flow shop scheduling problem with total tardiness criterion, which included four phases, i.e., discrete teaching phase, discrete probabilistic learning phase, population reconstruction, and neighborhood search. Ghazi and Ahiod [36] proposed an efficient TLBO approach for the discrete routing problem in wireless sensor networks, using a crossover technique, the Edge Recombination Operator to address the discrete representation of a solution. Recently, Wu et al. [20] presented a discrete hybrid TLBO algorithm to solve the discounted {0-1} knapsack problem, utilizing a double coding strategy (a quaternary and a real-numbered vector), self-learning factors and crossover operators to balance the exploitation and exploration abilities of the algorithm. Thus, at least to our knowledge, the TLBO algorithm has not been used for the solution of any VRP variant.

3 Prize-collecting vehicle routing problem

The mathematical formulation of the Prize-Collecting Vehicle Routing Problem defines that a predefined number of feasible routes M has to be constructed, utilizing the corresponding number of vehicles. Following the formulation presented in [12], the PCVRP can be described through a complete graph $Z = (V, A)$, where $V = \{0, \ldots, N\}$ is the set of nodes and $A = \{(i,j) | i,j \in V\}$ is the set of corresponding arcs.

Each node i included in the set $N_v = \{1, \dots, N\}$, represents a customer and, thus, specific values of prize, p_i and demand d_i are associated to it. The depot, i.e., the initial/final point, is denoted by node 0 and has zero prize and demand values. In addition, for each pair of nodes i, j, the traveling time between them can be expressed by their Euclidean distance, noted as c_{ij}, the symmetry of the problem defines that $c_{ij} = c_{ji}$. Furthermore, each vehicle has a maximum capacity of Q units of demand, and a large fixed usage cost G associated with it. Additionally, r denotes the task completion parameter (minimum demand ratio). Finally, as N_m is considered the set of nodes that are visited by vehicle m, $m = 1, \dots, M$. The following decision variables are used:

- $x_{ij}(i \neq j) = 1$ when node j is visited immediately after none i, otherwise $x_{ij} = 0$, $(i, j \in V)$.
- $y_i = 0$ when customer i is included in the solution, otherwise $y_i = 1$, $(i \in N_v)$.

The mathematical formulation of the PCVRP is:

$$Min : \sum_{i \in V} \sum_{j \in V, j \neq i} c_{ij} x_{ij} + G * M - \sum_{i=1}^{N_v} p_i (1 - y_i) \tag{1}$$

s.t.

$$\sum_{i=1}^{N} x_{i0} = \sum_{i=1}^{N} x_{0i} = M \tag{2}$$

$$\sum_{i=1}^{N} x_{ij} \leq 1, \quad j = 1, \dots, N \tag{3}$$

$$\sum_{j=1}^{N} x_{ij} \leq 1, \quad i = 1, \dots, N \tag{4}$$

$$\sum_{i \in N_m} d_i (1 - y_i) \leq Q, \quad m = 1, \dots, M \tag{5}$$

$$\sum_{i \in S} \sum_{j \in S, (j \neq i)} x_{ij} \leq |S| - 1, \quad \forall S \subset V \tag{6}$$

$$\frac{\sum_{i=1}^{N} d_i (1 - y_i)}{\sum_{i=1}^{N} d_i} \geq r \tag{7}$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V \tag{8}$$

$$y_i \in \{0, 1\}, \quad \forall i \in N_v \tag{9}$$

The goal of the objective function, Eq. (1), is the minimization of the total cost (traveled distance and fixed vehicle usage), by taking into account the total collected prize from the visited nodes. Constraint (2) requires that each vehicle conducts a route that initiates from the depot and returns to it. Constraints (3) and (4) ensure that each node is visited at most once, while Constraints (5) facilitate the capacity restrictions of the vehicle. Additionally, Constraints (6) are used to eliminate the sub-tours for each vehicle route. Constraint (7) ensures the minimum ratio of demand to be covered. Finally, Constraints (8) and (9) specify the integrity conditions on the variables.

4 Environmental prize-collecting vehicle routing problem

In the present paper, a novel VRP formulation is proposed, inspired by different widely studied green network design problems, to integrate an environmental aspect in the PCVRP, namely the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP). The main scope of the formulation is to translate the distance minimization part of the aforementioned objective function into a CO_2 emissions minimization objective.

In their recent survey, Dukkanci et al. [37] distinguish two VRP variations concerning the minimization of fuel consumption: the Energy Minimizing Vehicle Routing Problem (EMVRP) by Kara et al. [38], and the Pollution-Routing Problem (PRP) by Bektas, and Laporte [39]. The EMVRP is an extension of the classical VRP, where the objective is to minimize a distance-weighted load function in order to minimize the total energy consumption, while the PRP focuses on the minimization of a total cost function composed of labor, fuel, and emission costs expressed as a function of load, speed, and other parameters. Moreover, as Eskandarpour et al. [40] describe, researchers consider the minimization of fuel consumption as a function of distance and vehicle load throughout the route to address the energy consumption minimization. Xiao et al. [41] state that there is a general relationship between the Fuel Consumption Rate (FCR) and the vehicle's gross weight. FCR (ρ) was considered as a load (q) dependent function, applied to Capacitated VRP with an objective function to minimize fuel consumption, see Eq. (10), where ρ^* and ρ_0 denote the full-load and empty-load fuel consumption rate, respectively [39].

$$\rho(q) = \rho_0 + \frac{\rho^* - \rho_0}{Q} q \tag{10}$$

In 2014, Zhang et al. [42] proposed the Environmental Vehicle Routing Problem (EVRP) as a bi-objective problem, including the minimization of distance and CO_2 emissions. They assumed that the amount of CO_2 emissions emitted is a linear expression between the FCR and the weight of vehicle, see Eq. (11), where the CO_2 emissions rate is denoted by CER and considered as a relative fixed provided that the type of fuel is known, e.g., 2.61 kg/liter in case of the diesel oil.

$$e_{ijk} = CER * \left(\rho_0 + \frac{\rho^* - \rho_0}{Q} q_{ijk} \right) * c_{ij} \tag{11}$$

Moreover, Eglese and Bektas [43] described the current fuel consumption and emissions models found in the literature and how these models can be integrated into existing VRP formulations. However, the estimation of fuel consumption and pollutant emissions is a complicated task due to a variety of factors that are: travel-related (speed and acceleration rates), vehicle-related (engine size, fuel type, payload, and age of the vehicle), road-related (gradients, roundabouts, and traffic lights) and others. In the present research, only the vehicle load and the distance traveled are considered for the estimation of the CO_2 emissions. Precisely, the utilized emissions-function, of a Heavy-Duty Vehicle (HDV), can be found in Pan et al. [44] considering several assumptions, see Eq. (12). Other researchers have also, adopted this linear formulation of emission volume [45–48], for an HDV that has average speed of 80 km/h and fully loaded weights 25 tons, where:

- $E_{ij}(q,d)$: is the CO_2 emissions from a vehicle in kg/km with the variable of load q in ton and d in km,
- e_f: is the CO_2 emissions of a fully-loaded vehicle (1.096 kg/km for an HDV truck),
- e_e: is the CO_2 emissions of an empty vehicle (0.772 kg/km for an HDV truck) and
- q_{ij}: represents the aggregated volume of demand that the vehicle carries as it traverses from node i to node j.

$$E_{ij}(q,d) = c_{ij} * \left[\frac{(e_f - e_e)}{Q} (q_{ij}) + e_e \right] \tag{12}$$

Thus, the proposed mathematical formulation of the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP) emerges, by replacing the objective function, see Eq. (13), and expanding the decision variables for each vehicle/route m, i.e., x_{ij}^m. Also, a binary auxiliary decision variable y_{im} is used to define whether the node i is included in route m, where y_{im} equals to 1 if node i is included in route m and otherwise, it equals to zero. Moreover, the following constraints should be included to control the parameter q_{ij}^m, see Constraints (21), (22) and (23).

$$Min : \sum_{i \in V} \sum_{j \in V, j \neq i} \sum_{m=1}^{M} c_{ij} \left[\frac{(e_f - e_e)}{Q} q_{ij}^m + e_e x_{ij}^m \right] + G * M - \sum_{i=1}^{N} p_i (1 - y_{im}) \tag{13}$$

s.t.

$$\sum_{i=1}^{N} y_{i0} = \sum_{i=1}^{N} y_{0i} = M \tag{14}$$

$$\sum_{m=1}^{M} y_{im} \leq 1, \quad i = 1, \dots, N \tag{15}$$

$$\sum_{i=1}^{N} y_{im} \geq 1, \quad m = 1, \dots, M \tag{16}$$

$$\sum_{j=1, i \neq j}^{N} x_{ij}^{m} = y_{im}, \quad i = 1, \dots, N \quad \text{and} \quad m = 1, \dots, M \tag{17}$$

$$\sum_{i=1, i \neq j}^{N} x_{ij}^{m} = y_{jm}, \quad j = 1, \dots, N \quad \text{and} \quad m = 1, \dots, M \tag{18}$$

$$\sum_{i=1}^{N} y_{im} * d_i \leq Q, \quad m = 1, \dots, M \tag{19}$$

$$\frac{\sum_{i=1}^{N} \sum_{m=1}^{M} y_{im} * d_i}{\sum_{i=1}^{N} d_i} \geq r \tag{20}$$

$$\sum_{i=0, i \neq j}^{N} q_{ji}^{m} - \sum_{i=0, i \neq j}^{N} q_{ij}^{m} = d_j * y_{jm}, \quad j = 1, \dots, N \quad \text{and} \quad m = 1, \dots, M \tag{21}$$

$$0 \leq q_{ij}^{m} \leq Q * x_{ij}^{m}, \quad i,j = 1, \dots, N(i \neq j) \quad \text{and} \quad m = 1, \dots, M \tag{22}$$

$$q_{0i}^{m} = 0, \quad i = 1, \dots, N \quad \text{and} \quad m = 1, \dots, M \tag{23}$$

$$x_{ij}^{m} \in \{0,\ 1\}, \quad \forall i,j \in V, \quad \text{and} \quad m = 1, \dots, M \tag{24}$$

$$y_{im} \in \{0,\ 1\}, \quad \forall i \in N_v, \quad \text{and} \quad m = 1, \dots, M \tag{25}$$

Constraint (14) establish the correct number of routes to be formed. Constraints (15) guarantee that each node is included into a route no more than once and Constraints (16) ensure that each route should include at least one node. Constraints (17) and (18) ensure the continuity of the route. Constraints (19) and (20) are established to avoid the overload of a vehicle and ensure that the required ratio of demand is covered by the complete solution, respectively. Constraints (21) prohibits any illegal sub-tours and poses the continuity of the route in terms of changes in transferred demand volume. Constraints (22) control the feasible value range of the transferred demand volume, and Constraints (23) indicate that the vehicle starts empty-loaded from the depot node. Finally, Constraints (24) and (25) denote the range of the decision variables.

5 Teaching–learning-based optimization algorithm

Teaching–Learning-Based Optimization algorithm (TLBO) is a population-based heuristic stochastic optimization algorithm, proposed in 2011, by Rao et al. [7]. TLBO mimics the teaching and learning process of an ordinary classroom, where learners obtain knowledge not only from the teacher but also from their mutual interaction. In the TLBO algorithm, the learners of a class are regarded as the population, the different subjects are analogous to the decision variables in the optimization problem, and the teacher is defined as the best individual of the class, corresponding to the solution with the best fitness value. The standard TLBO algorithm entails a two-phase strategy, which consists of the Teacher-Phase and the Learner-Phase, respectively. The former phase simulates the teacher's effort to improve the average knowledge-level of the class; the latter phase is utilized to enhance each learner's performance by learning interactively from another randomly selected learner.

Thus, considering a D-dimensional optimization problem, each learner in a population of N_p individuals, at iteration t, is represented as a real-valued vector, $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)$. Learners are randomly initialized, with respect to the upper and lower boundaries of each decision variable. During the Teacher-Phase, the teacher, denoted by X_{tch}^t, aims to promote the average knowledge-level of the class. Hence, all learners are moved towards his/her knowledge-level, i.e., the position of the teacher, in the decision-variable space, taking into account the current mean knowledge-level of the class, denoted by X_{mean}^t. The position of each learner is updated following Eq. (26), where TF can be either 1 or 2, which is decided randomly with equal probability as $TF = round[1 + rand\{2-1\}]$, and $rand$ signifies a random floating number in the range [0, 1]. The new learner is accepted if the new corresponding fitness value is better than the old one.

$$X_{inew}^t = X_i^t + rand(X_{tch}^t - TFX_{mean}^t) \tag{26}$$

Subsequently, during the Learner-Phase, learners increase their knowledge-level through interaction between themselves. Such, for each learner i, another k $(i \neq k)$ is randomly selected. In case that the k selected learner is better than the current i, the latter is moved towards the former one, while otherwise, i learner is moved away k, following Eq. (27). The Learner-Phase is utilized to enhance the diversity of learners and to avoid premature convergence. Both phases are iteratively applied to the constructed population until the maximum number of iterations is reached. Algorithm 1 shows the pseudo-code of the standard TLBO.

$$X_{inew}^t = \begin{cases} X_i^t + rand(X_i^t - X_k^t), & \text{if } f(X_i^t) < f(X_k^t) \\ X_i^t + rand(X_k^t - X_i^t), & \text{otherwise} \end{cases} \tag{27}$$

Algorithm 1 Teaching-Learning-Based Optimization Algorithm

```
Define the objective function f(x);
Initialize and evaluate the learners' population X = X_1, ..., X_{N_p};
Find best learner: X_{tch} & Calculate mean level of the class X_{mean};
repeat
   for each X_i^t in N_p do
      % Teacher-Phase %
      Update learner i: X_{inew}^t = X_i^t + rand(X_{tch}^t − TF X_{mean}^t);
      if f(X_{inew}^t) < f(X_i^t) then
         X_i^t ← X_{inew}^t;
      end if
   end for
   % Learner-Phase %
   for each X_i in population do
      Select learner k (i ≠ k);
      if f(X_i^t) < f(X_k^t) then
         X_{inew}^t = X_i^t + rand(X_i^t − X_k^t)
      else
         X_{inew}^t = X_i^t + rand(X_k^t − X_i^t)
      end if
      if f(X_{inew}^t) < f(X_i^t) then
         X_i^t ← X_{inew}^t;
      end if
   end for
   Update teacher and mean;
   t ← t + 1;
until t > t_{max}
```

6 Proposed approach

TLBO algorithm is designed to solve continuous optimization problems, and such, it cannot be directly applied in the solution of the E-PCVRP since the respective solution should be expressed in a discrete way to represent a visit node-sequence. Other studies that can be found in the literature, which focus on the solution of combinatorial optimization problems using algorithms initially developed for continuous optimization, utilize techniques to encode and decode the solution. Specifically, the discrete-numbered solution vectors are mapped to the continuous space, the updating equations are applied, and subsequently, the vector is transformed back to discrete (or even binary) representation. For instance, such mechanisms entail the use of: the sigmoid function [49]; the hyperbolic function [50] and rounding after movement in the continuous space [51]. Additionally, a common approach is the hybridization of continuous algorithms with others that include discrete mechanisms. As mentioned above, the TLBO has been utilized in the solution of discrete problems, replacing the equations in Teacher-Phase and Learner-Phase by discrete operators. The present research focuses on a straightforward application of the standard TLBO scheme for solving the E-PCVRP, aiming to avoid transformation and approximation errors. Moreover, utilizing the proposed approach, the challenging task of designing proper discrete operators that correspond to the learning processes in the continuous domain is avoided.

Thus, in the present research, the "Coordinates Related" (CR) encoding/decoding process is utilized, which has been proposed by the authors in [52]. The CR is a problem-based method, and it holds the primary function of transforming a discrete-encoded solution to continuous values so that continuous updating mechanism can directly be applied in the solution of VRPs. Particularly, CR maps the solution that is represented as a sequence of nodes (discrete values) into a continuous domain, using the Cartesian coordinates of the respective nodes. In order for the reader to understand the proposed method, the solution representation of the E-PCVRP should be clarified. The widely used VRP solution representation found in the literature is a discrete-numbered vector [53]. The most relevant encoding mechanism to the presented research is the one of Goel, and Maini [54], according to which the start and the end of each route is node '0' and the visiting sequence of the rest of the available nodes is denoted by their index in the solution vector (X_i). Such as, an initial solution representation is depicted in the first row of Table 1. This node-sequence representation is suitable for calculating the objective function and is utilized for feasibility control.

Following the CR encoding process, for each solution vector in the respective population other two emerge, which include the corresponding x- and y-coordinate of the node positioned in a particular index. Consecutively, these new vectors include real-numbers, and their index-pair combination represents the node-sequence in the solution, accurately. Now, the learning equations of the TLBO can directly be applied in the two coordinates-vectors (Coor. x and Coor. y), resulting in two new ones (Coor. x-NEW and Coor. y-NEW). The incorporation of the CR process into the TLBO scheme imitates the movement of the visiting points in the 2-D. In order to identify the nodes that correspond to new visiting points (as defined by the new set of coordinates), the CR decoding process should be employed. According to the initially proposed CR process, the decoding is based on a randomly chosen direction. In more detail, the heuristic decoding process should investigate the search space near a new visiting point, moving either vertically or horizontally. In the present research, the decoding process is revised, and instead of a single-dimensional movement, the search of the feasible nodes that should correspond to the new visiting points is based on the Euclidean Distance. As an instance, according to the example presented in Table 1, where the discrete solution vectors are highlighted in bold, at index 2, the solution vector of the learner includes node 25, and the teacher's

Table 1 Example of the transformed vector according to CR

Position	1	2	3	4	5	6	7	8	9	10	11	12
X_{tch}	**0**	**2**	**14**	**22**	**32**	**18**	**3**	**0**	**31**	**17**	**8**	**0**
Coor. x	82	96	84	98	98	91	50	82	85	88	84	82
Coor. y	76	44	85	14	5	2	5	76	60	51	39	76
X_i	**0**	**25**	**19**	**10**	**12**	**5**	**3**	**0**	**13**	**30**	**0**	
Coor. x	82	61	19	2	5	13	50	82	98	20	82	
Coor. y	73	65	32	39	10	7	5	76	52	70	76	
Coor. x-NEW	82.01	61.02	18.99	3.21	5.12	13.42	49.89	82.35	98.31	20.05	82.10	
Coor. y-NEW	73.10	61.87	30.04	39.03	10.01	7.86	5.08	75.99	52.61	70.80	76.12	
X_{inew}	**0**	**15**	**19**	**8**	**12**	**5**	**3**	**0**	**13**	**30**	**0**	

vector includes node 2. Posterior to the learning mechanism, the corresponding new visiting point is located at (61.02, 61.87), and the algorithm seeks the closest feasible node of this point (from the available node-set), which is node 15 located at (61, 59), i.e., $\sqrt{(61-61.02)^2+(59-61.87)^2}=2.87$, in contrast to node 25, located originally at this index, i.e., $\sqrt{(61-61.02)^2+(65-61.87)^2}=3.13$.

Similarly, concerning index 3, the selected node 19 is located at (19, 32), closest to the new visiting point (18.99, 30.04) than any other considered node. In the case of equally-distant nodes, the one with the higher prize value should be selected. The above-described version of the CR is denoted by CRE. However, it is highly probable that the decoded updated solution will not be feasible. Thus, a simple heuristic should be employed that follows a two-phase procedure, based on the constraints of the problem at hand, namely the *FeasibilityControl*. In the first phase, the correct number (M) of constructed routes is imposed, starting from and terminating at node '0'. In the second phase, the nodes with the lowest prize value are removed from the solution, until the capacity of each route is under the predefined bound, while the minimum demand ratio of the solution should be respected.

Algorithm 2 TLBO-CRE

```
Define the objective function f(x);
Initialize and evaluate the learners' population (X = X_1, ..., X_{N_p}): Initial Population;
Find best learner: X_tch & Calculate mean level of the class X_mean;
repeat
  for each X_i^t in population do
    % Teacher-Phase %
    Employ CRE encoding process on X_i^t & X_tch^t ;
    Update learner i: X_inew^t = X_i^t + rand(X_tch^t − TF X_mean^t);
    Employ CRE decoding process on X_inew^t & Feasibility Control;
    Improve X_inew^t: 1 − 1 Exchange & Discard;
    if f(X_inew^t) < f(X_i^t) then
      X_i^t ← X_inew^t;
    end if
  end for
  % Learner-Phase %
  for each X_i in N_p do
    Select learner k (i ≠ k);
    Employ CRE encoding process on X_i^t & X_k^t ;
    if f(X_i^t) < f(X_k^t) then
      X_inew^t = X_i^t + rand(X_i^t − X_k^t)
    else
      X_inew^t = X_i^t + rand(X_k^t − X_i^t)
    end ifEmploy CRE decoding process on X_inew^t & Feasibility Control;
    Improve X_inew^t: 1 − 1 Exchange & Discard;
    if f(X_inew^t) < f(X_i^t) then
      X_i^t ← X_inew^t;
    end if
  end for
  Update teacher and mean;
  t ← t + 1;
until t > t_max
```

In Algorithm 2, the proposed TLBO-CRE is described. The proposed approach differentiates from the standard TLBO scheme, as the initial population is not randomly constructed, but it is generated using a heuristic approach (*Initial Population*). In addition, prior and after the update equations of Teacher-Phase and Learner-Phase, the CRE encoding/decoding process is utilized, accordingly, and local search techniques ($1-1$ *Exchange* and *Discard*) are employed to improve the quality of the updated solutions. The *Initial Population* is based on the savings algorithm published by Clarke and Wright in 1964 [55]. As described in Algorithm 3, the first step is the formulation of M initial routes that include only the depot and one other node. These routes are combined into an initial solution vector, and the non-included nodes are sorted based on their prize value. One-by-one, these nodes are positioned into the solution vector according to the savings method. That implies that the most efficient position (between nodes i, j) of a new node k to be inserted is where the following expression is minimized: $c_{i,k} + c_{k,j} - c_{i,j}$. Each route of the solution is augmented by nodes up to the predefined limit of capacity Q, while the demand reached from the complete solution should exceed the Q_{low} threshold (where $Q_{low} = r * \sum_{i=1}^{N} d_i$).

Algorithm 3 *Initial Population*

```
repeat
  Create M initial routes: [0 node 0];
  Combine the initial routes to vector;
  while Total capacity < Q_low do
    Create stack: sorted non-included nodes;
    for Each node k in stuck do
      Calculate the efficient position of node k;
      Correlate position to route m;
      if including node k in route m does not violate the capacity constraint Q then
        Include k and update the solution;
      end if
    end for
  end while
  Calculate the solutions' value in the objective function;
until Population N_p is constructed
```

Considering the utilized local search techniques included in both algorithmic phases of TLBO, the $1-1$ *Exchange* and the *Discard*, permutation, and removal operators are performed for a specific number of iterations, t_{local}. Following the $1-1$ *Exchange* repetitive procedure, two routes of the considered solution are randomly selected, and a position (a node) of one of the chosen routes is also randomly chosen. The second position needed for the exchange, i.e., a node from the second route is calculated using a distance vector. The end of the most expensive (in terms of distance) connection indicates the position of interest. The described node exchange aims to create a solution with less distance traveled (lower in cost), by omitting ineffective sequences of nodes and creating more efficient connections, as seen in Algorithm 4.

Algorithm 4 $1-1$ *Exchange*

repeat
  Randomly select routes: m_1 and m_2;
  Randomly select position in m_1, node: i;
  Compute distance vector of route m_2;
  Select position with the greatest distance value, node j;
  Compute total demand of both routes after the exchange;
  if New total demand values do not exceed the capacity limit **then**
    Calculate new distance traveled, exchanging the position of nodes i, j;
    Calculate the objective function of the new formation;
    if New objective function value is smaller than the initial one **then**
      Update the solution accordingly;
    end if
  end if
until t_{local} iterations reached

Under *Discard*, the algorithm attempts to remove nodes from the solution, reconstructing it, and consecutively, to achieve a solution of better quality without violating the minimum demand constraint. The adopted principle is that, when a node with low prize value is removed from the solution, the remaining solution is connected with a new arc, and in case that the new connection is efficient in terms of cost (distance), then, a better solution (with lower objective function value) could emerge. In order to enhance the diversification abilities of the proposed approach, the node to be removed from the solution vector is randomly chosen as the one with either the lowest prize value or the one with the highest (heavier) demand value; please see Algorithm 5.

Algorithm 5 *Discard*

repeat
  Randomly select route m;
  if $rand < 0.5$ **then**
    Find the node k with the smaller prize value in route m;
  else
    Find the node k with the highest demand value in route m;
  end if
  if Reducing the capacity of route m by d_k does not violate the constraints **then**
    Calculate new distance traveled, connecting the nodes before and after k;
    Calculate the objective function of the new formation;
    if New objective function value is smaller than the initial one **then**
      Update the solution accordingly;
    end if
  end if
until t_{local} iterations reached

7 Computational experiments

Since the Environmental-PCVRP is proposed in the present research, corresponding benchmark instances, along with their best-known solutions, are not to be found in the literature. Thus, in order to demonstrate the performance quality of the proposed TLBO-CRE, published benchmark instances of the standard PCVRP have been utilized. The related benchmark values are obtained from the solution of the proposed mathematical formulation of the E-PCVRP using a commercial solver, the Gurobi Optimizer. In addition, further comparisons are made utilizing other known bio-inspired algorithms, such as the constriction Particle Swarm Optimization (cPSO) algorithm [56], the Bat Algorithm (BA) [57] and two variants of the Differential Evolution (DE) algorithm [58]. Considering the DE solution approach, two different variants have been utilized, the DE/best/1/bin (employing binomial crossover) and the DE/best/1/exp (employing exponential crossover) [59], which in the following are denoted by DEbin and DEexp, respectively. For a fair comparison of the aforementioned algorithmic approaches, the CRE encoding/decoding process and the local search techniques described in Sect. 6 have been incorporated to all of them, considering equal population size. As the PCVRP has not been extensively studied, previous publications relied on the benchmark instances of the Capacitated Vehicle Routing Problem (CVRP). Recently, Long et al. [13] published a set of 120 instances for the PCVRP, accessible online, which is utilized in the present research. Specifically, for each of the 24 CVRP instances (group: A,B,E,M), 5 versions were generated by changing the ratio r : $\{0.60, 0.65, 0.70, 0.75, 0.80\}$. The different variants include problems with the number of nodes ranging from 32 to 200, and the number of vehicles from 4 to 17. Each instance is represented by a coding denoting the different topology, number of nodes, number of vehicles, and demand ratio, e.g., $A - n32 - k5 - 1$ indicates an instance of the group A with 32 nodes, 5 vehicles, and $r = 0.6$ (the last number in the decoding varies from 1 to 5, where 1 corresponds to $r = 0.6$, 2 corresponds to $r = 0.65$ and so on). The algorithmic computations were implemented in Matlab environment, and the solution of the described MIP, for comparison, was made using Gurobi 4.5.1 with Python 3.0 on an Intel(R) Core(TM) i7-7700HQ@2.80GHz with 8GB RAM.

As mentioned above, one of the main advantages of the standard TLBO algorithm is that it is an algorithm-specific parameter-less algorithm. Hence, only the size N_p and the maximum number of allowed t_{max} iterations should be considered. As, such a small parametric study has been conducted, and it has been concluded that the fittest values to promote the convergence rate of the proposed TLBO-CRE algorithm are $N_p = 40$ (from the range [10:100]) and $t_{max} = 80$, respectively. Moreover, since, in the proposed approach, local search techniques are included, the corresponding number of allows iterations has been set as $t_{local} = 400$. For a fair comparison between the proposed approach and the selected bio-inspired algorithms, their maximum computational time has been set at 60 s. Nevertheless, the algorithms selected for comparison require algorithm-specific parameter setting, and hence, following a parametric sampling search, the most promising values are reported in Table 2.

7.1 Comparative results with the mathematical solver (Gurobi Optimizer)

In Tables 3 and 4, the computational results on the tested benchmark instances of the mathematical solver (Gurobi Optimizer) and the proposed TLBO-CRE approach are presented. It should be noted that the maximum allowed solution time of the Gurobi Optimizer was fixed at 1500 s. The mathematical solver obtained a feasible solution on 90 instances, out of the 120 in total, within the specified time. At the same time, it was not able to prove the optimality of most of the obtained solutions, and thus, the objective value of the best-achieved solutions is reported. Moreover, in the aforementioned tables, for each instance, the percentage deviation pd_{min} based on the best solution that TLBO-CRE obtained, and the percentage deviation of the average objective value pd_{avg} over five algorithmic executions, are presented, to depict the performance deviation among the Gurobi Optimizer and the proposed approach. Thus, the reported positive percentage values correspond to TLBO-CRE solutions of better quality, while the negative ones demonstrate the lack of the proposed approach to outperform the mathematical solver. In total, TBLO-CRE outperformed the Gurobi Optimizer in 67 instances, including those unsolved by the latter approach, with an average pd_{min} equal to 1.275% (better quality) and – 0.397% (error). Additionally, the Gurobi Optimizer failed to achieve a feasible solution in any of the 20 considered instances of group M, which indicates the drawback of the mathematical solver in the solution of large and complex instances, e.g., with more than 101 nodes and ten routes, and highlights the necessity of a heuristic approach in the solution of real-world scenarios.

7.2 Comparative results with bio-inspired algorithms

For the comparison of the proposed TLBO-CRE approach with other bio-inspired algorithms, and thus, interpreting the reported values presented in Tables 5, 6, 7 and 8, a statistical analysis has been employed based on the best-obtained solution over

Table 2 Parameters of comparison algorithms

DEbin			DEexp		
Parameter	Sample range	Value	Parameter	Sample range	Value
F	[0.1, 1]	0.2	F	[0.1, 1]	0.7
Cr	[0.1, 1]	0.6	Cr	[0.1, 1]	0.5
BA			**cPSO**		
Parameter	Sample range	Value	Parameter	Sample range	Value
α	[0.90, 0.99]	0.94	c1	c1 + c2 = 4.1	2.7
γ	[0.90, 0.99]	0.99	c2		1.4
A_i^0		Random in [0.7, 1]			
r_i^0		Random in [0.1, 0.4]			

Table 3 Comparative results of TLBO-CRE and mathematical solver (Gurobi Optimizer) based on instance group A

Instances	Gurobi	TLBO-CRE	pd_{min} (%)	pd_{avg} (%)
A-n32-k5-1	4.309E+03	4.304E+03	0.100	0.008
A-n32-k5-2	4.408E+03	4.423E+03	– 0.330	– 0.444
A-n32-k5-3	4.297E+03	4.315E+03	– 0.405	– 0.480
A-n32-k5-4	4.329E+03	4.337E+03	– 0.188	– 0.256
A-n32-k5-5	4.606E+03	4.623E+03	– 0.374	– 0.443
A-n37-k6-1	5.385E+03	5.397E+03	– 0.213	– 0.442
A-n37-k6-2	5.415E+03	5.419E+03	– 0.067	– 0.121
A-n37-k6-3	5.344E+03	5.345E+03	– 0.031	– 0.097
A-n37-k6-4	5.390E+03	5.390E+03	– 0.012	– 0.119
A-n37-k6-5	5.109E+03	5.130E+03	– 0.426	– 0.690
A-n44-k6-1	5.011E+03	5.026E+03	– 0.314	– 0.372
A-n44-k6-2	4.832E+03	4.824E+03	0.166	– 0.077
A-n44-k6-3	5.220E+03	5.216E+03	0.076	– 0.206
A-n44-k6-4	4.777E+03	4.794E+03	– 0.355	– 0.459
A-n44-k6-5	4.995E+03	5.000E+03	– 0.117	– 0.202
A-n48-k7-1	5.869E+03	5.911E+03	– 0.711	– 0.881
A-n48-k7-2	6.137E+03	6.174E+03	– 0.607	– 0.805
A-n48-k7-3	6.027E+03	6.074E+03	– 0.787	– 0.901
A-n48-k7-4	6.015E+03	6.018E+03	– 0.052	– 0.350
A-n48-k7-5	5.968E+03	5.984E+03	– 0.269	– 0.355
A-n53-k7-1	5.910E+03	5.944E+03	– 0.576	– 0.702
A-n53-k7-2	5.653E+03	5.658E+03	– 0.093	– 0.225
A-n53-k7-3	5.684E+03	5.700E+03	– 0.267	– 0.436
A-n53-k7-4	5.483E+03	5.540E+03	– 1.046	– 1.167
A-n53-k7-5	5.890E+03	5.954E+03	– 1.094	– 1.184
A-n60-k9-1	7.793E+03	7.816E+03	– 0.286	– 0.366
A-n60-k9-2	7.819E+03	7.722E+03	1.243	1.106
A-n60-k9-3	8.157E+03	8.003E+03	1.881	1.705
A-n60-k9-4	7.719E+03	7.700E+03	0.252	– 0.031
A-n60-k9-5	8.026E+03	7.941E+03	1.060	0.752
A-n65-k9-1	7.634E+03	7.502E+03	1.722	1.628
A-n65-k9-2	7.378E+03	7.380E+03	– 0.024	– 0.221
A-n65-k9-3	7.523E+03	7.529E+03	– 0.089	– 0.183
A-n65-k9-4	7.412E+03	7.438E+03	– 0.359	– 0.668
A-n65-k9-5	7.253E+03	7.303E+03	– 0.684	– 1.001
A-n69-k9-1	7.541E+03	7.457E+03	1.119	0.989
A-n69-k9-2	7.374E+03	7.404E+03	– 0.411	– 0.543
A-n69-k9-3	7.345E+03	7.228E+03	1.589	1.463
A-n69-k9-4	7.177E+03	7.196E+03	– 0.260	– 0.404
A-n69-k9-5	7.318E+03	7.276E+03	0.577	0.257
A-n80-k10-3	8.708E+03	8.479E+03	2.625	2.591
A-n80-k10-5	8.836E+03	8.166E+03	7.584	7.398

Table 4 Comparative results of TLBO-CRE and mathematical solver (Gurobi Optimizer) based on instance groups B and E

Instances	Gurobi	TLBO-CRE	pd_{min} (%)	pd_{avg} (%)
B-n39-k5-1	3.967E+03	3.962E+03	0.122	0.041
B-n39-k5-2	4.085E+03	4.084E+03	0.024	− 0.112
B-n39-k5-3	3.911E+03	3.929E+03	− 0.473	− 0.642
B-n39-k5-4	4.018E+03	4.021E+03	− 0.095	− 0.203
B-n39-k5-5	3.932E+03	3.913E+03	0.474	0.371
B-n41-k6-1	5.076E+03	5.083E+03	− 0.146	− 0.199
B-n41-k6-2	5.037E+03	5.042E+03	− 0.089	− 0.420
B-n41-k6-3	4.896E+03	4.921E+03	− 0.516	− 0.679
B-n41-k6-4	4.798E+03	4.756E+03	0.864	0.531
B-n41-k6-5	4.957E+03	4.999E+03	− 0.841	− 1.132
B-n50-k7-1	5.996E+03	6.014E+03	− 0.286	− 0.344
B-n50-k7-2	5.477E+03	5.480E+03	− 0.043	− 0.161
B-n50-k7-3	5.669E+03	5.684E+03	− 0.255	− 0.405
B-n50-k7-4	5.707E+03	5.703E+03	0.072	− 0.034
B-n50-k7-5	5.765E+03	5.780E+03	− 0.262	− 0.379
B-n56-k7-1	5.265E+03	5.252E+03	0.231	0.109
B-n56-k7-2	5.267E+03	5.293E+03	− 0.488	− 0.671
B-n56-k7-3	5.237E+03	5.278E+03	− 0.787	− 0.912
B-n56-k7-4	5.435E+03	5.501E+03	− 1.211	− 1.308
B-n56-k7-5	5.413E+03	5.429E+03	− 0.297	− 0.327
B-n63-k10-1	8.773E+03	8.839E+03	− 0.754	− 0.836
B-n63-k10-2	8.552E+03	8.629E+03	− 0.907	− 0.995
B-n63-k10-3	8.959E+03	9.029E+03	− 0.781	− 1.022
B-n63-k10-4	8.787E+03	8.777E+03	0.118	− 0.144
B-n63-k10-5	9.103E+03	9.058E+03	0.499	0.299
B-n78-k10-1	8.001E+03	7.664E+03	4.223	4.165
B-n78-k10-3	8.239E+03	8.109E+03	1.578	1.334
B-n78-k10-4	8.051E+03	7.787E+03	3.280	2.968
E-n23-k3-1	2.583E+03	2.583E+03	0.000	0.000
E-n23-k3-2	2.646E+03	2.646E+03	0.000	0.000
E-n23-k3-3	2.652E+03	2.652E+03	0.000	0.000
E-n23-k3-4	2.467E+03	2.467E+03	0.000	0.000
E-n23-k3-5	2.487E+03	2.487E+03	0.000	0.000
E-n33-k4-1	3.715E+03	3.715E+03	0.000	0.000
E-n33-k4-2	3.457E+03	3.462E+03	− 0.164	− 0.355
E-n33-k4-3	3.323E+03	3.342E+03	− 0.555	− 0.673
E-n33-k4-4	3.340E+03	3.353E+03	− 0.374	− 0.557
E-n33-k4-5	3.304E+03	3.318E+03	− 0.411	− 0.573
E-n51-k5-1	3.336E+03	3.322E+03	0.421	0.063
E-n51-k5-2	3.485E+03	3.499E+03	− 0.373	− 0.530
E-n51-k5-3	3.422E+03	3.428E+03	− 0.182	− 0.318
E-n51-k5-4	3.421E+03	3.407E+03	0.408	0.216
E-n51-k5-5	3.504E+03	3.516E+03	− 0.330	− 0.552

Table 4 (continued)

Instances	Gurobi	TLBO-CRE	pd_{min} (%)	pd_{avg} (%)
E-n76-k10-1	8.233E+03	7.974E+03	3.147	3.047
E-n76-k10-2	8.034E+03	7.667E+03	4.578	4.464

five algorithmic execution and considering the solution quality performance of the examined algorithms on the four benchmark instance groups tested. Specifically, three non-parametric statistical tests for multiple comparisons have been selected and employed, i.e., Friedman, Friedman Aligned Ranks, and Quade tests [60]. Notably, these statistics can be used over real data, transforming them into ranks. The ranking can be achieved in different ways. The Friedman is based on n sets of ranks, one set for each instance (data-set) in our case, and the algorithms' performances are ranked separately for each one. The Friedman Aligned Ranks computes a value as the average performance achieved by all algorithms in each instance. Then, it calculates the difference between the performance obtained by an algorithm and this value (this step is repeated for all algorithms and instances). The resulting differences, called aligned observations, are used to obtain the final ranking. Also, the Friedman test considers all instances to be equal in terms of importance. In contrast, Quade test conducts a weighted ranking analysis, considering that some instances are more computationally complex. Thus, the rankings computed on each data set are scaled depending on the differences observed in the algorithms' performances. Moreover, in Fig. 1, the percentage deviation of each comparison algorithm from the results obtained by the proposed TLBO-CRE method (over the complete set of 120 instances) is presented via the related box-and-whisker diagrams. From these values' distribution, it can be seen that the comparison algorithms do not outperform the proposed approach. However, their competitive performance is also visible, as most of the percentage deviation values fall within a range close to zero, [− 0.05%, 0.05%]. Thus, the non-parametric statistical analysis is necessary to rank and evaluate the algorithms' performance.

The utilized statistical procedures consider that the null hypothesis being tested is that all algorithms obtain similar results with non-significant differences, considering a level of significance α. Table 9 presents the results of the non-parametric statistical analysis, which are the obtained ranking of the compared algorithms, the test statistic, and the p-value of each test, respectively. Interpreting these results is concluded that all three tests result in the same ranking, where the proposed TLBO-CRE is the best performing solution approach, with minimum rank values. Also, all three tests suggest the existence of significant performance differences among the algorithms considered, as all the computed statistics exceed their respective critical values. Specifically, considering $\alpha = 0.05$, for the statistics of Friedman test (Iman–Davenport extension) and Quade test, which are distributed according to an F distribution with $(5 - 1) = 4$ and $(5 - 1)(120 - 1) = 476$ degrees of freedom, the critical value is 2.39, and for the Friedman Aligned-Ranks test statistic, which is distributed according to an χ^2 distribution with $(5 - 1) = 4$ degrees of freedom, the critical value is 9.488. Finally, the computed p-values are less than α, and thus the

Table 5 Computational results of instance group: A

Instance	DEbin	DEexp	BA	cPSO	TLBO-CRE
A-n32-k5-1	4.313E+03	4.308E+03	4.314E+03	4.332E+03	4.304E+03
A-n32-k5-2	4.426E+03	4.429E+03	4.420E+03	4.440E+03	4.423E+03
A-n32-k5-3	4.315E+03	4.302E+03	4.321E+03	4.323E+03	4.315E+03
A-n32-k5-4	4.342E+03	4.343E+03	4.348E+03	4.342E+03	4.337E+03
A-n32-k5-5	4.662E+03	4.629E+03	4.618E+03	4.625E+03	4.623E+03
A-n37-k6-1	5.401E+03	5.403E+03	5.426E+03	5.410E+03	5.397E+03
A-n37-k6-2	5.413E+03	5.424E+03	5.433E+03	5.419E+03	5.419E+03
A-n37-k6-3	5.343E+03	5.336E+03	5.339E+03	5.346E+03	5.345E+03
A-n37-k6-4	5.385E+03	5.386E+03	5.391E+03	5.398E+03	5.390E+03
A-n37-k6-5	5.146E+03	5.149E+03	5.141E+03	5.141E+03	5.130E+03
A-n44-k6-1	5.020E+03	5.025E+03	5.029E+03	5.029E+03	5.026E+03
A-n44-k6-2	4.818E+03	4.821E+03	4.826E+03	4.835E+03	4.824E+03
A-n44-k6-3	5.222E+03	5.229E+03	5.227E+03	5.216E+03	5.216E+03
A-n44-k6-4	4.787E+03	4.803E+03	4.809E+03	4.796E+03	4.794E+03
A-n44-k6-5	4.998E+03	4.992E+03	5.006E+03	5.006E+03	5.000E+03
A-n48-k7-1	5.901E+03	5.908E+03	5.910E+03	5.916E+03	5.911E+03
A-n48-k7-2	6.153E+03	6.153E+03	6.147E+03	6.154E+03	6.174E+03
A-n48-k7-3	6.086E+03	6.084E+03	6.072E+03	6.070E+03	6.074E+03
A-n48-k7-4	6.034E+03	6.032E+03	6.085E+03	6.031E+03	6.018E+03
A-n48-k7-5	5.989E+03	5.964E+03	6.032E+03	5.982E+03	5.984E+03
A-n53-k7-1	5.929E+03	5.943E+03	5.953E+03	5.929E+03	5.944E+03
A-n53-k7-2	5.641E+03	5.641E+03	5.683E+03	5.635E+03	5.658E+03
A-n53-k7-3	5.695E+03	5.690E+03	5.641E+03	5.699E+03	5.700E+03
A-n53-k7-4	5.530E+03	5.503E+03	5.526E+03	5.531E+03	5.522E+03
A-n53-k7-5	5.940E+03	5.938E+03	5.527E+03	5.944E+03	5.954E+03
A-n60-k9-1	7.805E+03	7.809E+03	7.816E+03	7.830E+03	7.816E+03
A-n60-k9-2	7.740E+03	7.723E+03	7.758E+03	7.717E+03	7.722E+03
A-n60-k9-3	8.000E+03	7.986E+03	8.006E+03	7.975E+03	8.003E+03
A-n60-k9-4	7.723E+03	7.735E+03	7.721E+03	7.714E+03	7.700E+03
A-n60-k9-5	7.960E+03	7.977E+03	7.948E+03	7.959E+03	7.941E+03
A-n65-k9-1	7.496E+03	7.501E+03	7.479E+03	7.484E+03	7.502E+03
A-n65-k9-2	7.348E+03	7.349E+03	7.377E+03	7.384E+03	7.377E+03
A-n65-k9-3	7.526E+03	7.490E+03	7.521E+03	7.517E+03	7.506E+03
A-n65-k9-4	7.411E+03	7.447E+03	7.414E+03	7.430E+03	7.438E+03
A-n65-k9-5	7.298E+03	7.299E+03	7.270E+03	7.301E+03	7.303E+03
A-n69-k9-1	7.475E+03	7.475E+03	7.469E+03	7.461E+03	7.457E+03
A-n69-k9-2	7.401E+03	7.400E+03	7.418E+03	7.406E+03	7.404E+03
A-n69-k9-3	7.229E+03	7.222E+03	7.240E+03	7.227E+03	7.228E+03
A-n69-k9-4	7.203E+03	7.205E+03	7.191E+03	7.194E+03	7.196E+03
A-n69-k9-5	7.284E+03	7.295E+03	7.287E+03	7.286E+03	7.276E+03
A-n80-k10-1	8.538E+03	8.585E+03	8.596E+03	8.561E+03	8.589E+03
A-n80-k10-2	8.374E+03	8.350E+03	8.357E+03	8.355E+03	8.376E+03

Table 5 (continued)

Instance	DEbin	DEexp	BA	cPSO	TLBO-CRE
A-n80-k10-3	8.445E+03	8.449E+03	8.452E+03	8.464E+03	8.479E+03
A-n80-k10-4	8.664E+03	8.672E+03	8.630E+03	8.658E+03	8.597E+03
A-n80-k10-5	8.200E+03	8.132E+03	8.206E+03	8.149E+03	8.158E+03

Table 6 Computational results of instance group: B

Instance	DEbin	DEexp	BA	cPSO	TLBO-CRE
B-n39-k5-1	3.965E+03	3.964E+03	3.963E+03	3.963E+03	3.962E+03
B-n39-k5-2	4.091E+03	4.093E+03	4.093E+03	4.089E+03	4.084E+03
B-n39-k5-3	3.937E+03	3.941E+03	3.930E+03	3.931E+03	3.929E+03
B-n39-k5-4	4.024E+03	4.016E+03	4.027E+03	4.021E+03	4.021E+03
B-n39-k5-5	3.923E+03	3.921E+03	3.916E+03	3.914E+03	3.913E+03
B-n41-k6-1	5.068E+03	5.094E+03	5.094E+03	5.075E+03	5.083E+03
B-n41-k6-2	5.075E+03	5.060E+03	5.081E+03	5.065E+03	5.042E+03
B-n41-k6-3	4.919E+03	4.915E+03	4.925E+03	4.927E+03	4.921E+03
B-n41-k6-4	4.758E+03	4.768E+03	4.756E+03	4.779E+03	4.756E+03
B-n41-k6-5	4.980E+03	4.998E+03	5.011E+03	5.004E+03	4.999E+03
B-n50-k7-1	6.009E+03	6.010E+03	6.009E+03	6.011E+03	6.014E+03
B-n50-k7-2	5.472E+03	5.480E+03	5.473E+03	5.489E+03	5.480E+03
B-n50-k7-3	5.671E+03	5.678E+03	5.677E+03	5.683E+03	5.684E+03
B-n50-k7-4	5.708E+03	5.717E+03	5.710E+03	5.708E+03	5.703E+03
B-n50-k7-5	5.771E+03	5.771E+03	5.784E+03	5.789E+03	5.780E+03
B-n56-k7-1	5.260E+03	5.248E+03	5.243E+03	5.248E+03	5.252E+03
B-n56-k7-2	5.307E+03	5.279E+03	5.286E+03	5.291E+03	5.293E+03
B-n56-k7-3	5.274E+03	5.274E+03	5.278E+03	5.284E+03	5.278E+03
B-n56-k7-4	5.512E+03	5.504E+03	5.493E+03	5.495E+03	5.501E+03
B-n56-k7-5	5.425E+03	5.411E+03	5.416E+03	5.412E+03	5.426E+03
B-n63-k10-1	8.857E+03	8.832E+03	8.822E+03	8.868E+03	8.839E+03
B-n63-k10-2	8.659E+03	8.624E+03	8.647E+03	8.626E+03	8.629E+03
B-n63-k10-3	9.059E+03	9.033E+03	9.044E+03	9.044E+03	9.029E+03
B-n63-k10-4	8.827E+03	8.787E+03	8.798E+03	8.795E+03	8.777E+03
B-n63-k10-5	9.067E+03	9.056E+03	9.088E+03	9.048E+03	9.058E+03
B-n78-k10-1	7.758E+03	7.627E+03	7.644E+03	7.643E+03	7.618E+03
B-n78-k10-2	7.789E+03	7.737E+03	7.775E+03	7.760E+03	7.733E+03
B-n78-k10-3	8.249E+03	8.122E+03	8.114E+03	8.146E+03	8.109E+03
B-n78-k10-4	7.846E+03	7.786E+03	7.844E+03	7.839E+03	7.787E+03
B-n78-k10-5	8.129E+03	8.103E+03	8.042E+03	8.130E+03	8.054E+03

Table 7 Computational results of instance group: E

Instance	DEbin	DEexp	BA	cPSO	TLBO-CRE
E-n23-k3-1	2.583E+03	2.583E+03	2.583E+03	2.583E+03	2.583E+03
E-n23-k3-2	2.646E+03	2.646E+03	2.646E+03	2.646E+03	2.646E+03
E-n23-k3-3	2.652E+03	2.652E+03	2.652E+03	2.652E+03	2.652E+03
E-n23-k3-4	2.467E+03	2.467E+03	2.467E+03	2.467E+03	2.467E+03
E-n23-k3-5	2.487E+03	2.487E+03	2.487E+03	2.487E+03	2.487E+03
E-n33-k4-1	3.716E+03	3.715E+03	3.715E+03	3.715E+03	3.715E+03
E-n33-k4-2	3.469E+03	3.464E+03	3.466E+03	3.463E+03	3.462E+03
E-n33-k4-3	3.347E+03	3.343E+03	3.338E+03	3.341E+03	3.342E+03
E-n33-k4-4	3.367E+03	3.354E+03	3.353E+03	3.366E+03	3.353E+03
E-n33-k4-5	3.325E+03	3.321E+03	3.321E+03	3.319E+03	3.318E+03
E-n51-k5-1	3.374E+03	3.344E+03	3.359E+03	3.359E+03	3.322E+03
E-n51-k5-2	3.513E+03	3.499E+03	3.504E+03	3.499E+03	3.499E+03
E-n51-k5-3	3.447E+03	3.417E+03	3.428E+03	3.431E+03	3.428E+03
E-n51-k5-4	3.430E+03	3.408E+03	3.413E+03	3.417E+03	3.407E+03
E-n51-k5-5	3.539E+03	3.519E+03	3.519E+03	3.513E+03	3.516E+03
E-n76-k10-1	8.027E+03	7.972E+03	7.983E+03	7.978E+03	7.974E+03
E-n76-k10-2	7.718E+03	7.666E+03	7.661E+03	7.656E+03	7.667E+03
E-n76-k10-3	7.699E+03	7.669E+03	7.661E+03	7.670E+03	7.668E+03
E-n76-k10-4	7.780E+03	7.758E+03	7.767E+03	7.759E+03	7.757E+03
E-n76-k10-5	7.883E+03	7.807E+03	7.810E+03	7.804E+03	7.810E+03
E-n101-k14-1	1.086E+04	1.082E+04	1.081E+04	1.082E+04	1.081E+04
E-n101-k14-2	1.131E+04	1.125E+04	1.124E+04	1.123E+04	1.125E+04
E-n101-k14-3	1.102E+04	1.094E+04	1.095E+04	1.094E+04	1.095E+04
E-n101-k14-4	1.085E+04	1.079E+04	1.079E+04	1.078E+04	1.078E+04
E-n101-k14-5	1.153E+04	1.127E+04	1.127E+04	1.127E+04	1.127E+04

null hypothesis is rejected at a high level of significance. It should be noted that according to the Quade test ranking, the rank values of the proposed approach and the DEexp are almost equal.

8 Conclusions

The present research introduces the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP) as a new VRP variant that extends the original selective routing problem (i.e., the PCVRP) since the variable cost/ distance minimization part of the objective function is replaced by a load-distance function for CO_2 emissions minimization. The corresponding mathematical formulation of the E-PCVRP is presented, which includes new decision variables and constraints, comparing to the original PCVRP formulation found in the literature. Considering the solution of the E-PCVRP, an algorithmic approach based on the Teaching–Learning-Based

Table 8 Computational results of instance group: M

Instance	DEbin	DEexp	BA	cPSO	TLBO-CRE
M-n101-k10-1	6.940E+03	6.922E+03	6.923E+03	6.922E+03	6.927E+03
M-n101-k10-2	7.106E+03	7.069E+03	7.089E+03	7.081E+03	7.077E+03
M-n101-k10-3	7.007E+03	6.970E+03	6.991E+03	6.956E+03	6.964E+03
M-n101-k10-4	7.008E+03	6.878E+03	6.917E+03	6.903E+03	6.894E+03
M-n101-k10-5	6.979E+03	6.943E+03	6.967E+03	6.939E+03	6.934E+03
M-n121-k7-1	3.212E+03	2.979E+03	3.008E+03	2.955E+03	2.973E+03
M-n121-k7-2	3.457E+03	3.260E+03	3.330E+03	3.307E+03	3.270E+03
M-n121-k7-3	3.392E+03	3.207E+03	3.165E+03	3.224E+03	3.201E+03
M-n121-k7-4	3.074E+03	2.965E+03	3.045E+03	3.010E+03	2.973E+03
M-n121-k7-5	3.286E+03	3.187E+03	3.210E+03	3.222E+03	3.187E+03
M-n151-k12-1	7.480E+03	7.379E+03	7.386E+03	7.395E+03	7.374E+03
M-n151-k12-2	7.057E+03	6.906E+03	6.934E+03	6.952E+03	6.926E+03
M-n151-k12-3	7.296E+03	7.170E+03	7.228E+03	7.243E+03	7.150E+03
M-n151-k12-4	6.937E+03	6.866E+03	6.840E+03	6.852E+03	6.821E+03
M-n151-k12-5	6.716E+03	6.606E+03	6.600E+03	6.635E+03	6.597E+03
M-n200-k17-1	1.074E+04	1.049E+04	1.046E+04	1.050E+04	1.046E+04
M-n200-k17-2	1.104E+04	1.076E+04	1.076E+04	1.077E+04	1.075E+04
M-n200-k17-3	1.050E+04	1.025E+04	1.031E+04	1.032E+04	1.030E+04
M-n200-k17-4	1.047E+04	1.013E+04	1.017E+04	1.019E+04	1.017E+04
M-n200-k17-5	1.054E+04	1.027E+04	1.033E+04	1.033E+04	1.033E+04

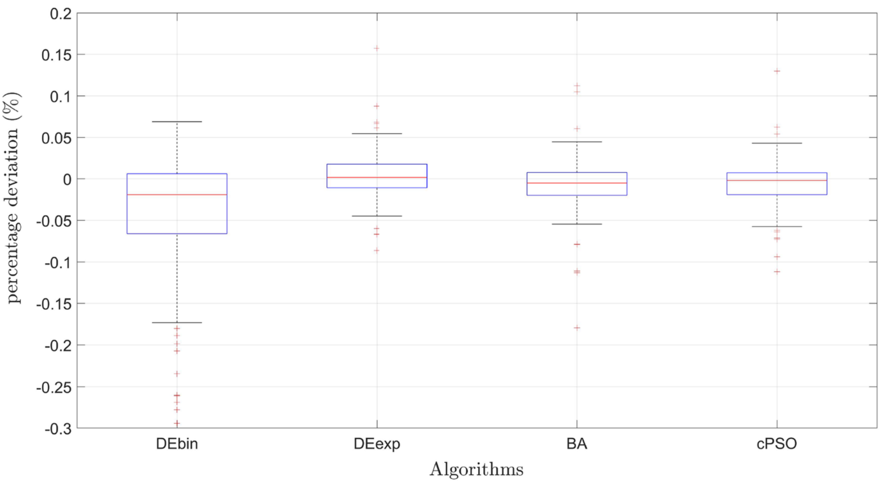

Fig. 1 Percentage deviation of comparison algorithms from the TLBO-CRE results

Optimization algorithm is proposed, namely the TLBO-CRE. Particularly, a novel encoding/decoding process has been presented and utilized, namely the CRE, to

Table 9 Non-parametric statistical analysis over the 120 benchmark instances

	Algorithm	Friedman test	Friedman aligned-ranks test	Quade test
Ranking	TLBO-CRE	2.479	235.570	2.392
	DEbin	3.708	421.983	4.047
	DEexp	2.6	244.683	2.397
	BA	3.2	307.050	3.157
	cPSO	3.0125	293.212	3.005
Statistic F		12.829	81.228	19.329
p-value		6.318E−10	1.110E−16	9.436E−15

make feasible the employment of the TLBO standard algorithmic scheme to the solution of the VRP problem at hand. The learning mechanisms incorporated in the TLBO, are based on equations, which are applied to solution vectors that contained continuous values for each decision variable. However, the E-PCVRP requires a discrete-numbered solution representation to represent the visiting sequence of the selected nodes. Consecutively, the CRE encoding process is applied, and each solution is correlated to two new continuous-valued vectors that consist of the x- and y-coordinates of each included node, respectively. Thus, the learning equations of the TLBO are directly applied to these vectors, that are updated accordingly. In order to evaluate the new solution, the CRE decoding process is utilized, and each one of the new pairs of coordinates corresponds to a node based on its proximity in the 2-D space (Euclidean Distance). As such, the E-PCVRP is optimized using the TLBO-CRE, without applying hybridization or discretization techniques in the standard algorithmic framework. The effectiveness of the proposed TLBO-CRE framework has been proven over computational experiments in the solution of benchmark instances found in the literature. For these computational experiments, a commercial mathematical solver, the Gurobi Optimizer, and other bio-inspired algorithms, i.e., DE, BA, cPSO, have been utilized. Based on the conducted statistical analysis, TLBO-CRE outperforms all the aforementioned solution approaches. To further explore the capabilities of the TLBO-CRE, and since the CRE could be tailored to fit any VRP, the proposed approach could also be tested in the solution of other problems. As an instance, for future research, the optimization of the PCVRP-NP is proposed, where the number of routes to be constructed is not a pre-specified value, in a multi-objective formulation, along with the corresponding extension of the E-PCVRP.

Funding This research is co-financed by Greece and the European Union (European Social Fund- ESF) through the Operational Programme: Human Resources Development, Education and Lifelong Learning, in the context of the project "Strengthening Human Resources Research Potential via Doctorate Research" (MIS-5000432), implemented by the State Scholarships Foundation (IKY) (Nos. 2018-050-0502-13964, 2018-050-0502-13953).

References

1. European Environment Agency, Greenhouse gas emissions from transport in Europe, Indicator Assessment. https://www.eea.europa.eu/data-and-maps/indicators/transport-emissions-of-greenhouse-gases/transport-emissions-of-greenhouse-gases-12. Accessed 20 March 2020
2. European Commission, White paper roadmap to a single European transport area towards a competitive and resource efficient transport system. https://ec.europa.eu/transport/themes/strategies/2011_white_paper_en. Accessed 20 March 2020
3. Scaburi, A., Ferreira, J.C., Steiner, M.T.A.: Sustainable logistics: a case study of vehicle routing with environmental considerations. In: International Business, Trade and Institutional Sustainability, pp. 765–779. Springer (2020)
4. Demir, E., Huckle, K., Syntetos, A., Lahy, A., Wilson, M.: Vehicle routing problem: past and future. In: Wells, P. (ed.) Contemporary Operations and Logistics. Palgrave Macmillan, Cham (2019)
5. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications. SIAM, Philadelphia (2014)
6. Tang, L., Wang, X.: Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem. Int. J. Adv. Manuf. Technol. **29**, 1246–1258 (2006)
7. Rao, R.V., Savsani, V.J., Vakharia, D., Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design **43**(3), 303–315 (2011)
8. Osaba, E., Yang, X.S., Del Ser, J.: Is the vehicle routing problem dead? An overview through bioinspired perspective and a prospect of opportunities. In: Yang, X.S., Zhao, Y.X. (eds.) Nature-Inspired Computation in Navigation and Routing Problems. Springer Tracts in Nature-Inspired Computing. Springer, Singapore (2020)
9. Zhang, T., Chaovalitwongse, W.A., Zhang, Y.J., Pardalos, P.M.: The hot-rolling batch scheduling method based on the prize collecting vehicle routing problem. J. Ind. Manag. Optim. **5**(4), 749–765 (2009)
10. Jia, S.J., Yi, J., Yang, G.K., Du, B., Zhu, J.: A multi-objective optimization algorithm for the hot rolling batch scheduling problem. Int. J. Prod. Res. **51**(3), 667–681 (2013)
11. Tiwari, A., Chang, P.C., Elangovan, G., Annadurai, S.P.: A hybrid edge recombination approach to solve price collecting vehicle routing problem. In: 2015 International Conference on Control, Automation and Robotics, pp. 200–203. IEEE (2015)
12. Li, K., Tian, H.: A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. Appl. Soft Comput. **43**, 469–479 (2016)
13. Long, J., Sun, Z., Pardalos, P.M., Hong, Y., Zhang, S., Li, C.: A hybrid multi-objective genetic local search algorithm for the prize-collecting vehicle routing problem. Inf. Sci. **478**, 40–61 (2019)
14. Stenger, A., Schneider, M., Goeke, D.: The prize-collecting vehicle routing problem with single and multiple depots and non-linear cost. EURO J. Transp. Logist. **2**, 57–87 (2013)
15. Bulhões, T., Ha, M.H., Martinelli, R., Vidal, T.: The vehicle routing problem with service level constraints. Eur. J. Oper. Res. **265**(2), 544–558 (2018)
16. Orlis, C., Laganá, D., Dullaert, W., Vigo, D.: Distribution with quality of service considerations: the capacitated routing problem with profits and service level requirements. Omega **93**, 102034 (2019)
17. Zou, F., Chen, D., Xu, Q.: A survey of teaching-learning-based optimization. Neurocomputing **335**, 366–383 (2019)
18. Nayak, J., Naik, B., Chandrasekhar, G.T., Behera, H.S.: A survey on teaching-learning-based optimization algorithm: short journey from 2011 to 2017. In: Computational Intelligence in Data Mining, pp. 739–758. Springer, Singapore (2019)
19. Xue, R., Wu, Z.: A Survey of Application and Classification on Teaching-learning-based Optimization Algorithm. IEEE Access (2019)
20. Wu, C., Zhao, J., Feng, Y., Lee, M.: Solving discounted 0–1 knapsack problems by a discrete hybrid teaching-learning-based optimization algorithm. Appl. Intell. **50**, 1–17 (2020)
21. Kashyap, A.K., Pandey, A.: Optimized path planning for three-wheeled autonomous robot using teaching-learning-based optimization technique. In: Advances in Materials and Manufacturing Engineering, pp. 49–57. Springer, Singapore (2020)
22. Dede, T.: Application of teaching-learning-based-optimization algorithm for the discrete optimization of truss structures. KSCE J. Civ. Eng. **18**, 1759–1767 (2014)

23. Toğan, V., Eirgash, M.A.: Time-cost trade-off optimization of construction projects using teaching learning based optimization. KSCE J. Civ. Eng. **23**, 10–20 (2019)
24. Naik, A.: Use of teaching learning based optimization for data clustering. In: Sharma, R., Mishra, M., Nayak, J., Naik, B., Pelusi, D. (eds.) Innovation in Electrical Power Engineering, Communication, and Computing Technology. Lecture Notes in Electrical Engineering, vol. 630. Springer, Singapore (2020)
25. Xu, Y., Yang, Z., Li, X., Kang, H., Yang, X.: Dynamic opposite learning enhanced teaching-learning-based optimization. Knowledge-Based Systems **188**, 104966 (2020)
26. Shukla, A.K., Singh, P., Vardhan, M.: An adaptive inertia weight teaching-learning-based optimization algorithm and its applications. Appl. Math. Model. **77**, 309–326 (2020)
27. Mishra, R., Sharma, N., Sharma, H.: Half-life teaching factor based TLBO algorithm. In: Advances in Data and Information Sciences, pp. 257–269. Springer, Singapore (2020)
28. Peng, Z.K., Zhang, S.X., Zheng, S.Y., Long, Y.L.: Collective information-based teaching-learning-based optimization for global optimization. Soft Comput. **23**(22), 11851–11866 (2019)
29. Zhou, G., Moayedi, H., Foong, L.K.: Teaching-learning-based metaheuristic scheme for modifying neural computing in appraising energy performance of building. Eng. Comput. **36**, 1–12 (2020)
30. Deb, S., Gao, X.Z., Tammi, K., Kalita, K., Mahanta, P.: A new teaching-learning-based chicken swarm optimization algorithm. Soft Comput. **24**, 1–19 (2019)
31. Li, J.Q., Pan, Q.K., Mao, K.: A discrete teaching-learning-based optimisation algorithm for realistic flowshop rescheduling problems. Eng. Appl. Artif. Intell. **37**, 279–292 (2015)
32. Lotfipour, A., Afrakhte, H.: A discrete teaching-learning-based optimization algorithm to solve distribution system reconfiguration in presence of distributed generation. Int. J. Electr. Power Energy Syst. **82**, 264–273 (2016)
33. Chen, D., Zou, F., Lu, R., Yu, L., Li, Z., Wang, J.: Multi-objective optimization of community detection using discrete teaching-learning-based optimization with decomposition. Inf. Sci. **369**, 402–418 (2016)
34. Shao, W., Pi, D., Shao, Z.: A hybrid discrete optimization algorithm based on teaching-probabilistic learning mechanism for no-wait flow shop scheduling. Knowledge-Based Syst. **107**, 219–234 (2016)
35. Shao, W., Pi, D., Shao, Z.: A hybrid discrete teaching-learning based meta-heuristic for solving no-idle flow shop scheduling problem with total tardiness criterion. Comput. Oper. Res. **94**, 89–105 (2018)
36. El Ghazi, A., Ahiod, B.: Energy efficient teaching-learning-based optimization for the discrete routing problem in wireless sensor networks. Appl. Intell. **48**(9), 2755–2769 (2018)
37. Dukkanci, O., Bektaş, T., Kara, B.Y.: Green network design problems. In: Sustainable Transportation and Smart Logistics, pp. 169–206. Elsevier (2019)
38. Kara, I., Kara, B.Y., Yetis, M.K.: Energy minimizing vehicle routing problem. In: International Conference on Combinatorial Optimization and Applications, pp. 62–71. Springer, Berlin (2007)
39. Bektaş, T., Laporte, G.: The pollution-routing problem. Transp. Res. B Methodol. **45**(8), 1232–1250 (2011)
40. Eskandarpour, M., Ouelhadj, D., Fletcher, G.: Decision making using metaheuristic optimization methods in sustainable transportation. In: Sustainable Transportation and Smart Logistics, pp. 285–304. Elsevier (2019)
41. Xiao, Y., Zhao, Q., Kaku, I., Xu, Y.: Development of a fuel consumption optimization model for the capacitated vehicle routing problem. Comput. Oper. Res. **39**(7), 1419–1431 (2012)
42. Zhang, S., Lee, C.K.M., Choy, K.L., Ho, W., Ip, W.H.: Design and development of a hybrid artificial bee colony algorithm for the environmental vehicle routing problem. Transp. Res. D Transp. Environ. **31**, 85–99 (2014)
43. Eglese, R., Bekta, S.T.: Chapter 15: Green vehicle routing. In: Vehicle Routing: Problems, Methods, and Applications, 2nd edn, pp. 437–458. Society for Industrial and Applied Mathematics (2014)
44. Pan, F., Ye, C., Wang, K., Cao, J.: Research on the vehicle routing problem with time windows using firefly algorithm. JCP **8**(9), 2256–2261 (2013)
45. El Bouzekri, A., Elhassania, M.E., Alaoui, A.E.: A hybrid ant colony system for green capacitated vehicle routing problem in sustainable transport. J. Theor. Appl. Inf. Technol. **54**, 198–208 (2013)
46. Ayadi, R., ElIdrissi, A. E., Benadada, Y., Alaoui, A. E. H.: Evolutionary algorithm for a green vehicle routing problem with multiple trips. In: 2014 International Conference on Logistics Operations Management, pp. 148–154. IEEE (2014)

47. Wu, C.Y., Visutarrom, T., Chiang, T.C.: Green vehicle routing problem: the tradeoff between travel distance and carbon emissions. In: 2018 15th International Conference on Control, Automation, Robotics and Vision, pp. 1659–1664. IEEE (2018)
48. Karagul, K., Sahin, Y., Aydemir, E., Oral, A.: A simulated annealing algorithm based solution method for a green vehicle routing problem with fuel consumption. In: Lean and Green Supply Chain Management, pp. 161–187. Springer, Cham (2019)
49. Sayadi, M., Ramezanian, R., Ghaffari-Nasab, N.: A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. Int. J. Ind. Eng. Comput. **1**(1), 1–10 (2010)
50. Chandrasekaran, K., Simon, S.P., Padhy, N.P.: Binary real coded firefly algorithm for solving unit commitment problem. Inf. Sci. **249**, 67–84 (2013)
51. Baghlani, A., Makiabadi, M.H., Sarcheshmehpour, M.: Discrete optimum design of truss structures by an improved firefly algorithm. Adv. Struct. Eng. **17**(10), 1517–1530 (2014)
52. Trachanatzi, D., Rigakis, M., Taxidou, A., Marinaki, M., Marinakis, Y., Matsatsinis, N.: A novel solution encoding in the differential evolution algorithm for optimizing tourist trip design problems. In: International Conference on Learning and Intelligent Optimization, pp. 253–267. Springer, Cham (2019)
53. Li, J., Li, T., Yu, Y., Zhang, Z., Pardalos, P.M., Zhang, Y., Ma, Y.: Discrete firefly algorithm with compound neighborhoods for asymmetric multi-depot vehicle routing problem in the maintenance of farm machinery. Appl. Soft Comput. **81**, 105460 (2019)
54. Goel, R., Maini, R.: A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems. J. Comput. Sci. **25**, 28–37 (2018)
55. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
56. Clerc, M., Kennedy, J.: The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. **6**(1), 58–73 (2002)
57. Yang, X.S.: A New Metaheuristic Bat-Inspired Algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Studies in Computational Intelligence, vol. 284. Springer, Berlin (2010)
58. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
59. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artif. Intell. Rev. **33**(1–2), 61–106 (2010)
60. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**(1), 3–18 (2011)

Authors and Affiliations

Dimitra Trachanatzi[1] · Manousos Rigakis[1] · Magdalene Marinaki[1] · Yannis Marinakis[1]

Manousos Rigakis
mrigakis@isc.tuc.gr

Magdalene Marinaki
magda@dssl.tuc.gr

Yannis Marinakis
marinakis@ergasya.tuc.gr

[1] Technical University of Crete, School of Production Engineering and Management University Campus, Chania, Crete, Greece